

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日  
Date of Application:

2003年 5月20日

出 願 番 号  
Application Number:

特願2003-141292

[ST.10/C]:

[JP2003-141292]

出 願 人  
Applicant(s):

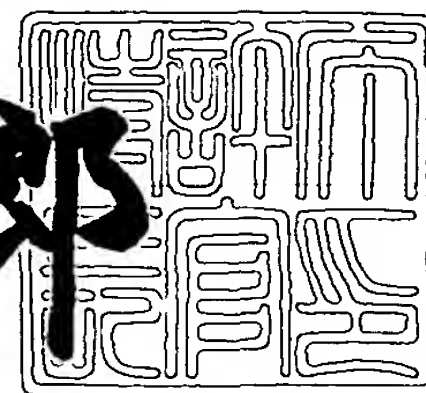
株式会社日立製作所

U.S. App'n Filed 8-22-03  
Inventor. Y. Miki et al  
Mattingly Stanger & Major  
Docket H-1110

2003年 6月20日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

太田信一郎



出証番号 出証特2003-3048564

【書類名】 特許願

【整理番号】 H02012811A

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 13/00

【発明者】

    【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所中央研究所内

    【氏名】 三木 良雄

【発明者】

    【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所中央研究所内

    【氏名】 水野 和彦

【特許出願人】

    【識別番号】 000005108

    【氏名又は名称】 株式会社 日立製作所

【代理人】

    【識別番号】 100075096

    【弁理士】

    【氏名又は名称】 作田 康夫

    【電話番号】 03-3212-1111

【手数料の表示】

    【予納台帳番号】 013088

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 J O B 分配制御方法

【特許請求の範囲】

【請求項 1】

複数のサーバをそれぞれ備えた複数の計算センタがネットワークを介して接続された計算機システムにおいて、第 1 の計算センタでの J O B 要求の発生に対し該第 1 の計算センタから第 2 の計算センタへの当該 J O B の投入を判断する J O B 分配制御方法であって、

前記第 1 の計算センタと前記第 2 の計算センタとを含むシステムでの所望する J O B レスポンスタイムを予め設定すること、

前記第 1 の計算センタでの平均的な J O B 要求間隔と各サーバでの J O B 平均処理時間を入力として前記所望する J O B レスポンスタイムの達成に必要なサーバ数を算出し、前記第 1 のサーバについてのサーバ数の基準値と比較すること、

必要なサーバ数が前記サーバ数の基準値を越えたときに、前記第 2 の計算機センタへ J O B を投入することを特徴とする J O B 分配制御方法。

【請求項 2】

ネットワークによって複数の計算機センタが接続された計算機システムの J O B 分配制御方法であって、

第 1 の計算機センタから第 2 の計算機センタに J O B を投入する際に、

平均的な J O B の要求間隔と、平均的な J O B の実行時間と、所望する J O B のレスポンスタイムとから、実行に必要なサーバ数を算出する手段を設け、

必要なサーバ数が第 1 の計算機センタにおいて予め定められた基準値を越えたときに、第 2 の計算機センタへ J O B を投入し、

第 2 の計算機センタは利用者の認証を行い、投入された J O B の実行時間に対して課金することを特徴とする J O B 制御方法。

【請求項 3】

ネットワークによって複数の計算機センタが接続された計算機システムにおいて、

第 1 の計算機センタから第 2 の計算機センタに J O B を投入する際に、

平均的なJOBの要求間隔と、平均的なJOBの実行時間と、所望するJOBのレスポンスタイムとから、実行に必要なサーバ数を算出する手段を設け、

必要なサーバ数が第1の計算機センタにおいて予め定められた基準値を越えたときに、第2の計算機センタへJOBを投入し、

第2の計算機センタは第1のセンタから投入されるJOBの平均的な投入間隔と、平均的なJOBの実行時間と、所望するJOBのレスポンスタイムとから、実行に必要なサーバ数を算出する手段を持ち、

必要なサーバ数が第2の計算機センタにおいて予め定められた基準値を越えたときに、第3の計算機センタへJOBを投入することを特徴とするJOB制御方法。

【請求項4】

グリッド等のようにネットワークによって複数の計算機センタが接続された計算機システムにおいて、

第1の計算機センタから第2の計算機センタにJOBを投入する際に、

平均的なJOBの要求間隔と、平均的なJOBの実行時間と、所望するJOBのレスポンスタイムとから、実行に必要なサーバ数を算出する手段を設け、

必要なサーバ数が第1の計算機センタにおいて予め定められた基準値を越えたときに、第2の計算機センタへJOBを投入し、

第2の計算機センタは第1の計算機センタから投入されたJOBに対して利用料金の算出をする手段を設け、第2の計算機センタを利用した分の料金を第1の計算機センタへ請求することを特徴とする、計算機センタの課金方法。

【請求項5】

グリッド等のようにネットワークによって複数の計算機センタが接続された計算機システムにおいて、

第1の計算機センタから第2の計算機センタにJOBを投入する際に、

平均的なJOBの要求間隔と、平均的なJOBの実行時間と、所望するJOBのレスポンスタイムとから、実行に必要なサーバ数を算出する手段を設け、

必要なサーバ数が第1の計算機センタにおいて予め定められた基準値を越えたときに、第2の計算機センタへJOBを投入し、

第2の計算機センタは第1の計算機センタと第2の計算機センタの利用比率を

計算する手段を設け、第 2 の計算機センタの利用比率が予め設定された基準値を超えた場合に、計算機を第 1 の計算機センタへ販売することを特徴とする計算機販売方法。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は計算機を運用する計算機センタで、複数の計算機にJOB実行を分配する方法に関し、特にグリッドやインターネットのような広域ネットワークで接続された複数の計算機センタ間での連携を前提としたJOB制御ソフトウェア技術に係わる。

【 0 0 0 2 】

【従来の技術】

複数の計算機を所有する計算機センタでは、JOBを実行する際に各計算機の負荷がなるべく均等になるようにJOBを分配する。この分配は負荷分散技術と呼ばれ、インターネットに接続された計算機センタであるデータセンタ向けの負荷分散装置により実現される。負荷分散装置の技術は例えば非特許文献 1 に示されているように、実行要求のあったJOBを利用可能な計算機に均等に配分するラウンドロビン方式やIPハッシング方式が知られている。この方式はJOBがほぼ同じ処理時間で終了することが想定される場合には、負荷分散方法として有効である。しかし、どれだけの処理時間要求があるのか不明なJOBを取り扱う際には、各計算機の負荷を観測してJOBを分散する必要がある。特許文献 1 にそのような技術が開示されており、JOB実行中の計算機負荷を観測し、負荷の少ない計算機にJOBを投入する。計算機がグリッドやインターネット等のネットワークにより広域に分散し接続され存在している場合には、ラウンドロビンあるいはIPハッシングにより、負荷分散すると、利用する可能性のある全ての計算機が均等に利用される。また、負荷を観測する方法では、ネットワークを介して計算機の負荷を問い合わせ、その負荷情報に基づいてJOBの分配を制御する。

【特許文献 1】

特開平 5 - 2 6 5 9 5 5 号公報

## 【非特許文献 1】

<http://online.plathome.co.jp/products/clikarray/index.phtml>

## 【 0 0 0 3 】

## 【発明が解決しようとする課題】

上述のように、ラウンドロビンやIPハッシングでは使用する可能性のある計算機全てが、JOB分配の対象となる。ところが、ネットワークを介して複数の計算機センタが接続されている場合には、JOBの発生箇所に近い計算機センタと遠隔に位置する計算機センタの計算機とを使い分ける方が便利である。具体的には、JOB実行の発生した計算センタにおいて計算機の処理能力に余裕がある限りはその計算機センタにてJOBを実行し、余力が無い場合には遠隔のセンタにてJOBを実行することが望ましい。ところが、JOBを均等に分配する方法では、JOBの実行要求が発生した計算機センタに余力が存在しても、遠隔の計算機センタにJOBが分配されることが発生する。

## 【 0 0 0 4 】

この問題を回避するために負荷を観測する方法を採用したとしても、この方法ではグリッドやネットワークを介して負荷を観測するために、ネットワークの遅延時間が負荷の観測時間に含まれてしまい、計算機負荷の誤認識が発生する。つまり、遠隔のセンタにJOBを投入する場合には、投入を決定してから実際にJOBが実行されるまでの時間と、計算機の負荷を観測して遠隔センタから報告するのにそれぞれネットワークの時間がかかり、JOBが実行される直前に負荷を観測してしまい、軽負荷であると誤認識することがあり得る。このように、従来技術ではグリッド等によってネットワークを介し広域に分散して計算機が配置されていることが配慮されていない。

## 【 0 0 0 5 】

そこで、本発明の課題は、グリッドのようにネットワークで接続され広域に分散した計算機センタの計算機の負荷分散、JOB分配に関して、JOBの発生した計算機センタに余力のある場合にはそのセンタでJOBを実行し、遠隔の計算機センタを利用する場合においてもネットワークの遅延時間に影響されず、均等にJOBを分配する方法を与えることにある。



## 【 0 0 0 6 】

## 【課題を解決するための手段】

上記の課題は次の手段により解決される。JOB実行の要求が発生する計算機センタをローカルセンタ、ネットワークにより接続された遠隔の計算機センタをリモートセンタとする。所望のJOBレスポンスタイムを事前に設定し、JOB投入時間の平均値、JOB実行時間の平均値を入力とし、先のレスポンスタイムを実現するために必要なサーバ数を決定する手段を有し、必要サーバ数がローカルセンタの基準値を越えたときに、リモートセンタへJOB実行することにより、計算機の負荷を実際に観測すること無しに適正なJOB分配を実現する。 【発明の実施の形態】

図1に本発明の1実施形態である計算機システムの動作フローを示す。また、図3にはその計算機システムの概略構成を示す。

## 【 0 0 0 7 】

図3においてローカルセンタ300はJOB要求が発生する計算機センタであり、センタ内にはサーバ310aおよび310bを所有している。このサーバ310aまたは310bまたはその両方には後に詳述するようにJOB管理機構311がプログラムとして実装されており、本発明を実施するに必要なJOBリクエストキュー、JOBリザルトキュー、JOB要求観測、サーバ数決定などの機能を実現する。リモートセンタ301a、および301bはローカルセンタ300とネットワーク303を介して接続された計算機センタであり、ローカルセンタ300とリモートセンタ301a、301bは物理的に広域に分散して配備されていることを前提とする。リモートセンタ301aにもローカルセンタ300と同様に、サーバ310c、310dが設置されており、ここでもいずれかのサーバ上のプログラムとしてJOB管理機構312aが実現される。リモートセンタ301aでは、JOBリザルトキューの他にサーバ負荷の観測手段やJOBの課金管理の仕組みが実装される。リモートセンタ301bもリモートセンタ301aと同じくサーバ310e、310fを有しており、同じくJOB管理機構312bを具備している。 JOB管理機構311、312aあるいは312bによって実現される本発明のJOB制御方法を図1に戻り説明する。処理手順全体は初期パラメータ設定100とJOB実行サービス110に大別され、初期パラメータ設定100ではローカルセンタとリモートセンタとの通常用いる

サーバ台数、JOBの要求があってからJOBが終了する時間、すなわちJOBレスポンスタイム等の標準的な値を設定する処理である。より具体的には、JOB要求間隔の平均時間であるJOB標準間隔を設定し(101)、サーバ上でJOBが実行開始し、終了するまでの標準時間であるJOB実行標準時間を設定する(102)。また、前述のJOBレスポンスタイムをローカルセンタとリモートセンタからなる全システムの望ましい時間、つまり設計値として設定する(103)。以上の設定値から、後述するサーバ数計算方法によって、全システムで必要なサーバ数の標準値を決定する(104)。

#### 【 0 0 0 8 】

以上、初期パラメータ設定処理100によって、全システムの標準的な挙動に係る数値が決定され、次のJOB実行サービス110によってJOB要求が無くなるまでの、半永久的に継続する全システムの動作が定義される。まずローカルセンタにおいて、JOB実行の要求が発生すると(111)、最も最近要求のあった時刻との差を計算し、JOB間隔の平均時間を計算する(112)。図面には明記していないが、平均時間は次の二つの方法によって計算可能である。一つはJOB要求を観測する時間間隔を別途設定し、その間に発生したJOB要求の平均時間を計算するものである。この方法では、JOB要求間隔に変化があった場合にその変化を迅速に摘出することが可能である。別法としては、JOB要求間隔時間の総和を求め、JOB要求数で割ることにより平均値を求めることができる。この方法は、容易に実現可能である反面、JOB要求間隔に変化があっても検出しにくい欠点がある。

#### 【 0 0 0 9 】

JOB要求の間隔時間の平均値が計算できた時点で、先に述べた初期パラメータ設定100の結果と合わせ、システム全体として設定されたJOBレスポンスタイムとサーバ上でJOBが実行される平均時間が揃うことになる。そこで、後述する理論式により、設定されたJOBレスポンスタイムを実現するのに必要なサーバ数を計算する(113)。この必要サーバ数がローカルセンタに対して予め定めた基準値より大きい場合にのみ(114)、リモートセンタのサーバにJOBを投入する(117)。リモートセンタでは、投入されたJOBを実行し(118)、JOBの結果をローカルセンタへ返送する(119)。



## 【 0 0 1 0 】

必要サーバ数の判定(114)における基準値としては、ローカルセンタで利用可能なサーバ数を用いるのが基本である。この他にも、サーバ上で多重実行可能なJOBの数を設定し、多重実行JOB数とサーバ数との積を基準値とする方法等が可能である。なお、必要サーバ数がローカルセンタの基準値に収まる場合(114)はローカルセンタのサーバにJOBが投入され(115)、JOBが実行される(116)。つまり、ローカルセンタとリモートセンタとから成る全システムに関して、所望のJOBレスポンスタイムを実現するために、ローカルセンタのサーバで十分な限り、ローカルセンタにおいてJOBを実行するのが本発明の本質的な部分である。この基本的処理は外部からセンタの運用に関して停止要求があるまで継続される(120)。

## 【 0 0 1 1 】

以上が本発明の骨子であるが、図2を用いてリモートセンタのJOB実行に関わる別の実施形態を述べる。図2のJOB実行サービス210においては、リモートセンタのサーバにJOBが投入された後に、利用者の認証処理(212)を実施する。具体的には、リモートセンタのサーバにJOBを投入する際に、ローカルセンタでJOBを要求したユーザのID情報をJOBと同時にリモートセンタに伝送する。ユーザのIDは例えばUNIX（登録商標）オペレーティングシステムのパスワードファイルにて管理されるIDを用いることが可能である。また、本発明のように、計算センタ同士が自動的にJOBをやり取りする場合においては、リモートセンタにおいて、ローカルセンタのユーザ全てを管理することは煩雑であることが懸念される。この場合には、利用認証212においてユーザIDの代わりにグループIDを代用することが可能である。グループIDも先に述べたUNIX（登録商標）オペレーティングシステムのグループIDを使うことが可能である。このように、認証過程を設けることで、ユーザ、またはグループといった利用者単位にリモートセンタ利用に関する課金を行うことが出来る(214)。この図2に示した発明の実施形態では、上記に説明したリモートセンタへのJOB投入211、利用認証212、課金処理214以外は図1に示した実施形態の各処理と同様の動作となる。したがって、図2に示した実施形態では、JOBを実行しようとした場合に、ローカルセンタ内のサーバだけで所望のレスポンスタイムでJOBが終了出来ない場合には費用が発生する。

## 【 0 0 1 2 】

次にもう一つの実施形態について図4のシステム構成図並びに図5のフローチャートを用いて説明する。図3ではJOB要求を発生するローカルセンタ300が図1または図2に示した方法によって、サーバ数を決定し、リモートセンタ301aまたは301bにJOBを投入する。図4では、JOB要求を発生するローカルセンタ400に対して、JOBを投入された第1のリモートセンタ401がローカルセンタ400と同様な方法で必要サーバ数を計算し、第2のリモートセンタ402へJOBを投入するシステム構成を示している。この場合、第1のリモートセンタ401のJOB管理機構にはローカルセンタ400のJOB管理機構410同様の仕組みが必要となる。具体的な第1リモートセンタにおけるJOB制御方法を図5に示す。初期パラメータ設定510はローカルセンタ400と全く同一の処理内容となる。ここでのJOB標準間隔、JOB実行標準時間、JOBレスポンスタイムはローカルセンタ400の各設定値と同一であることが望ましいが、異なる値であっても発明の本質は変化しない。JOB実行サービス520では、ローカルセンタで実施されるJOB制御と同様の処理が第1リモートセンタ401として実行される。すなわち、ローカルセンタからJOBが投入される(521)と、そのJOB投入間隔の平均値を計算し(522)、初期パラメータ設定510で決定したJOB投入の標準時間間隔とサーバ上で実行される平均実行時間、および所望のJOBレスポンスタイムから必要サーバ数を算出し、必要サーバ数が第1のリモートセンタの基準値を越えた場合に(524)、第2のリモートセンタのサーバへJOBを投入する。もし、必要サーバ数が第1のリモートセンタの基準値を下回る場合には第1のリモートセンタのサーバにJOBを投入し(530)、JOBを実行する(531)。このように複数のリモートセンタが関与する場合には、課金管理は必須であり、第2リモートセンタでは第1リモートセンタから伝送されたユーザ情報に基づき利用者認証526を実施し、JOB実行527後に課金処理528を行い、実行結果を第1リモートセンタへ返送する(529)。

## 【 0 0 1 3 】

次に図6を用いて図1や図3におけるJOB管理機構とサーバとの関係を詳細に説明する。サーバ310aはローカルセンタに設置されたサーバであり、サーバ310cはリモートセンタに設置されたサーバである。それぞれのサーバには、プログラ

ムを実行する c C P U 610a, 610b とメモリ 611a, 611b およびネットワークインターフェース 619a, 619b が具備されている。メモリ 611a, 611b はプログラム領域とデータ領域とに分割されており、図 1 に示した JOB 管理機構はプログラム領域のクライアント JOB 管理プログラム 612 (ローカルセンタ用)、サーバ JOB 管理プログラム 613 (リモートセンタ用) として格納されている。図 1 のローカルセンタのサーバに JOB を投入する場合 (115)、リモートセンタのサーバに JOB を投入する場合 (117) のいずれの場合においても、サーバ上で動作中の JOB の数は動作 JOB 数 616a, 616b として管理されており、動作 JOB 数が非零の場合またはある基準値を越えている場合には、JOB を即時実行できないとみなし、JOB を JOB リクエストキュー 614 に JOB を積み、後述するような JOB をキューから解放する条件が整うまで JOB の実行を待つ。また、図 1 JOB 間隔時間平均値計算 112 では、過去に発生した JOB 要求の時刻と、JOB 要求間隔の総和を記憶しておく必要があり、そのためにデータ領域には最新 JOB 到着時刻 618 の領域が確保される。実行が終了した JOB はまずリモートセンタの JOB リザルトキュー 615b に積まれ、ローカルセンタが結果を受け取れる状態になるとすぐに転送される。図 2 の利用認証 212 ではリモートセンタを利用可能なユーザまたはグループの認証が必要であり、そのパスワードなどの認証に必要な情報がユーザリスト 617 として格納される。

#### 【 0 0 1 4 】

上記した JOB リクエストのキューも含めた JOB 制御方法の詳細フローを図 7 に示す。ここで、JOB 実行サービス 710 においてリモートセンタのサーバに JOB が投入された場合 (711)、実行中の JOB の数を動作 JOB 数 616 b から取得し、この値がリモートセンタにおけるサーバ数を越えているか否かを判定する (712)。越えている場合には、即時実行不可能であるので、JOB を JOB リクエストキュー 614 に積む (711a)。ローカルセンタのサーバに JOB が投入される場合 (709) も、リモートセンタの場合と同様に、実行中の JOB の数を動作 JOB 数 616a から取得し、この値がローカルセンタにおけるサーバ数を越えているかを判定し (713)、こえる場合には JOB を JOB リクエストキュー 614 に積む (711b)。

#### 【 0 0 1 5 】

JOB リクエストキュー 614 から JOB を取りだし、実行する時の動作を図 8 に示す

。図 7 で JOB キューイングとして示した処理では、常に JOB リクエストキューが空でないか否かをチェックし(800)、空でないときには、JOB が終了したサーバがある場合を検出し(801)、終了した JOB がある場合にはキューから JOB を取り出してサーバで実行する(802)。この処理はキューが空でない限り繰り返される。

【 0 0 1 6 】

ここまでの説明が本発明における基本的な JOB 制御方法であり、次に図 1 に代表されるサーバ数 113 の処理方法を理論的背景および前提条件を含めて説明する。まず、JOB の要求はランダムに発生すると考える。これは、待ち行列理論で述べるランダム到着を意味する。また、JOB がサーバ上で実行される際の時間もランダムに変化すると考える。また、JOB がキューの中に存在する時間とサーバで実行されている時間の和をレスポンスタイムと定義する。さらに、サーバを待ち行列理論の処理窓口と考え、サーバ数つまり窓口の数を  $s$  と定義する。以上のモデル化したイメージを図 9 に示す。ここでさらに、JOB の要求間隔の平均時間を  $1/\lambda$ 、サーバで実行される実行される JOB の平均処理時間を  $1/\mu$  とし、キューの長さ  $n$  である確立を  $p_n$  とすると、 $s$  個の窓口を持つ待ち行列理論から次式が成立する。

【 0 0 1 7 】

【数 1】

$$\begin{aligned} \mu p_1 &= \lambda p_0 && \dots\dots\dots \text{式(a)} \\ \lambda p_{n-1} + (n+1)\mu p_{n+1} &= (\lambda + n\mu)p_n && (1 \leq n \leq s-1) \quad \dots\dots\dots \text{式(b)} \\ \lambda p_{n-1} + s\mu p_{n+1} &= (\lambda + s\mu)p_n && (n \geq s) \quad \dots\dots\dots \text{式(c)} \end{aligned} \quad \text{(数1)}$$

【 0 0 1 8 】

ここで、JOB 要求と処理時間の比として  $a = \lambda / \mu$ 、および系の安定性の指標として  $\rho = a/s$  を定義すると、(数 1) の式(a)と式(b)から次式が導かれる。

【 0 0 1 9 】

【数 2】

$$\begin{aligned} p_1 &= ap_0 \\ p_n &= \frac{a^n}{n!} p_0 && (0 \leq n \leq s) \end{aligned} \quad \text{(数2)}$$

【 0 0 2 0 】

一方、 $n \geq s$  のときには、(数 1) の式(c)を  $p_n$  の隣接項について再整理した次式が成り立つ。

【 0 0 2 1 】

【数 3】

$$s\mu(p_{n+1}-p_n) = \lambda(p_n - p_{n-1}) \quad (\text{数3})$$

【 0 0 2 2 】

さらに、 $n=s$  の境界条件から(数 4) が成立しているため、(数 5) が得られる。

【 0 0 2 3 】

【数 4】

$$s\mu p_s = \lambda p_{s-1} \quad (\text{数4})$$

【 0 0 2 4 】

【数 5】

$$p_n = \frac{a^n}{s! s^{n-s}} p_0 = \frac{s^s}{s!} \frac{\rho^n}{s!} p_0 \quad (\text{数5})$$

【 0 0 2 5 】

なお、 $p_0$  については  $\sum p_n = 1$  から下記のようになる。

【 0 0 2 6 】

【数 6】

$$p_0 = \frac{1}{\sum_{n=0}^{s-1} \frac{a^n}{n!} + \frac{a^s}{(s-1)!(s-a)}} \quad (\text{数6})$$

【 0 0 2 7 】

以上から $p_n$ の完全解が得られ、これによりモデル化したローカルセンタとリモートセンタに関して、下記の諸指標を得ることができる。

JOBリクエストキューの長さ $L_q$ ：

【 0 0 2 8 】

【数 7】

$$L_q = \sum_{n=1}^{\infty} n p_n = \frac{\lambda \mu a^s}{(s-1)! (s\mu - \lambda)^2} \quad (\text{数7})$$

【 0 0 2 9 】

JOBリクエストキュー内での待ち時間 $W_q$ ：

【 0 0 3 0 】

【数 8】

$$W_q = L_q \frac{1}{\lambda} = \frac{\mu a^s}{(s-1)! (s\mu - \lambda)^2} \quad (\text{数8})$$

【 0 0 3 1 】

JOBのレスポンスタイム $W$ ：

【 0 0 3 2 】

【数 9】

$$W = W_q + \frac{1}{\mu} \quad (\text{数9})$$

【 0 0 3 3 】

ここまでの式の導出により、全システムの平衡状態においてJOBのレスポンスタイムがJOBの要求時間間隔 $1/\lambda$ の関数として表現できたことになる。そこで、JOBの要求状況が変化しても一定のレスポンスタイムを保持するために必要な窓口数（サーバ数）をもとめることができる。

【 0 0 3 4 】



【数 1 0】

$$\frac{dW(\lambda)}{d\lambda} = \frac{d}{d\lambda} \left[ \frac{\mu a^s}{(s-1)! (s\mu - \lambda)^2} + \frac{1}{\mu} \right] = 0 \quad (\text{数10})$$

【0 0 3 5】

この方程式は  $(p_0)' = 0$  を仮定することにより  $s$  に関して解くことができる。したがって、入力JOBの頻度変化に対してレスポンスタイムを不変とする為に必要なサーバ数  $s$  は次式によって与えられる。

【0 0 3 6】

【数 1 1】

$$s = \frac{a + \sqrt{a^2 - 8a}}{2} \quad (\text{数11})$$

【0 0 3 7】

以上によって、図1の初期パラメータ設定100によって設定されたJOBレスポンスタイムを不変とするために必要なサーバ数を平均的なJOB要求間隔とJOBの実行時間とから求めることができる。

【0 0 3 8】

最後に本発明を用いたビジネスの実施形態を図10を用いて述べる。図中中央の2本の縦線はローカルセンタとリモートセンタにおける処理の時系列を表している。ローカルセンタでは、図2に示したように、標準的なレスポンスタイムを設定し(1000)、標準的なJOB要求時間間隔を設定し(1001)、標準的なJOB実行時間を設定する(1002)ことで、ローカルセンタの標準的な値が設定される(1003)。その後、JOB要求の受付が開始してJOB要求がある毎に必要なサーバ数の決定がなされる(1005)。次にサーバ数がローカルセンタサーバの基準値を越えた場合(1006)にリモートセンタのサーバを利用する(1007)。ここで、リモートセンタではユーザ毎あるいはローカルセンタ毎にリモートサーバの利用時間の合計を計算し(1008)し、その総計に対して利用料金の請求を行う(1010)。これにより、ローカルセンタは利用料金を支払い(1009)、リモートセンタは利益を得る(1011)。これによ

り、ローカルセンタに備えられた計算機資源の能力を越えたJOBをリモートセンタの支援によって実行可能とすることができ、それにより、リモートセンタはビジネスを行うことが可能である。

【 0 0 3 9 】

図 1 1 はビジネスの別の実施形態を示したものである。図において、リモートセンタはローカルセンタ、リモートセンタ双方の利用時間総計を計算し(1101)、その比率を明らかにする。このとき、リモートセンタの利用比率が予め定められた基準値よりも高い時に(1102)、リモートセンタはローカルセンタにサーバを販売する(1103)。これにより、ローカルセンタはサーバ料金として金銭をリモートセンタに支払い(1104)、リモートセンタは利益を得る(1105)ことができる。

【 0 0 4 0 】

【発明の効果】

本発明によれば、リモートセンタの負荷情報を観測せずにJOBの割り当て、分配を決定するために、ローカルセンタのJOB処理能力を越えた場合にのみリモートセンタを利用することが可能であり、ネットワークに遅延時間が存在しても、負荷を観測せずにJOBの分配を行うために、負荷の誤認識を回避したJOB分配が可能である。

【図面の簡単な説明】

【図 1】

本発明の 1 実施形態の動作を示すフローチャートである。

【図 2】

JOB実行に課金処理が伴う別の実施形態の動作を示すフローチャートである。

【図 3】

図 1 の実施形態のシステム構成を示すブロック図である。

【図 4】

更に別の実施形態のシステム構成を示すブロック図である。

【図 5】

図 4 の実施形態の動作を示すフローチャートである。

【図 6】

図 1 の実施形態もしくは図 2 の実施形態のサーバ内の詳細な構成を示すブロック図である。

【図 7】

JOBキューイングの動作をも含めた実施形態の詳細動作を示すフローチャートである。

【図 8】

キューの制御方法を示すフローチャートである。

【図 9】

本発明の原理を示すための待ち行列モデルである。

【図 1 0】

本発明によるビジネス形態を示すタイムシーケンス図である。

【図 1 1】

本発明によるビジネス形態の変形例を示すタイムシーケンス図である。

【符号の説明】

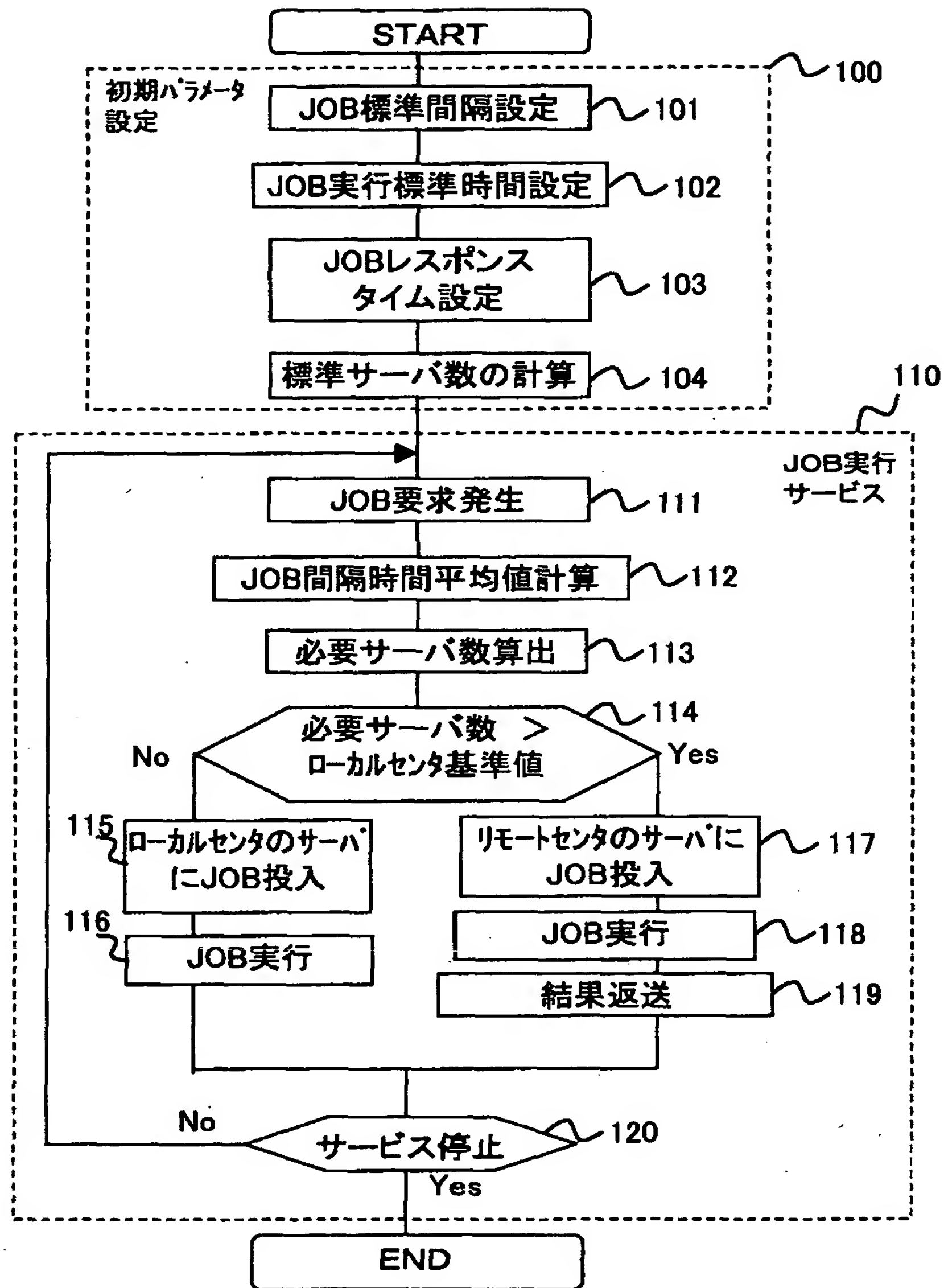
- 100 : パラメータ設定段階
- 110 : JOB実行サービス段階
- 300 : ローカルセンタ
- 301a,301b : リモートセンタ
- 303 : ネットワーク
- 310a～310f : サーバ
- 311,312a,312b : JOB管理機構。

【書類名】

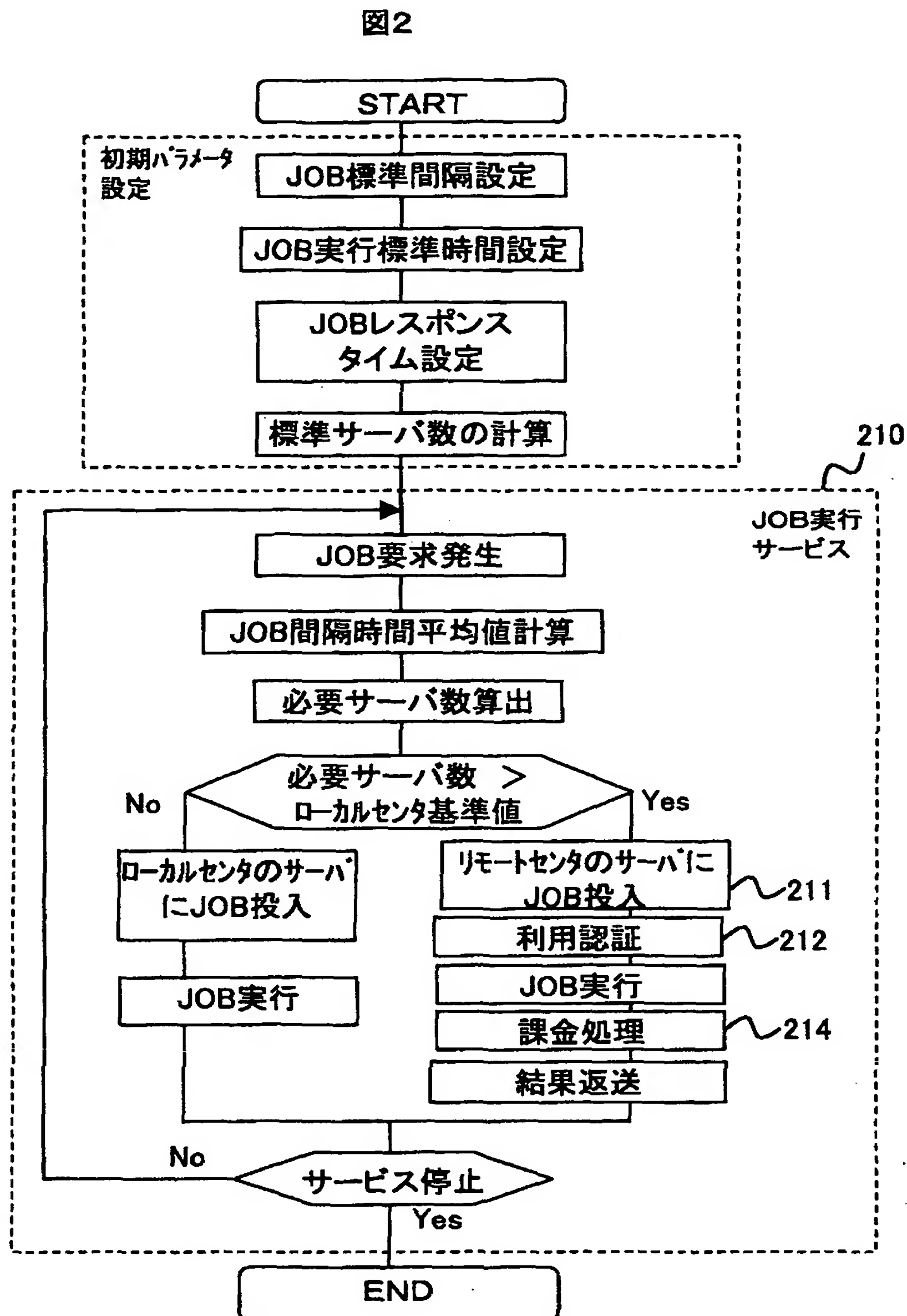
図面

【図 1】

図1

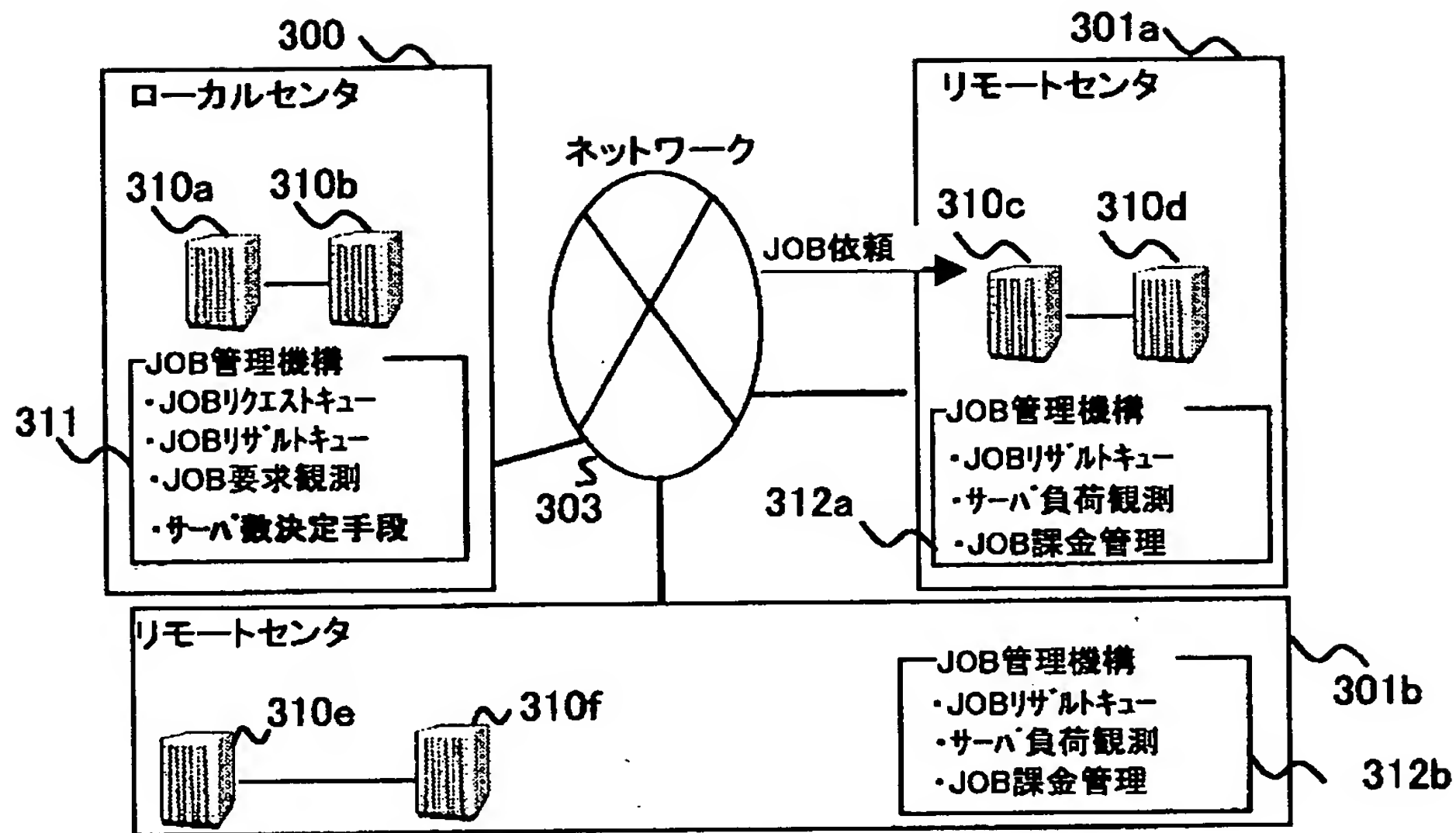


【図2】



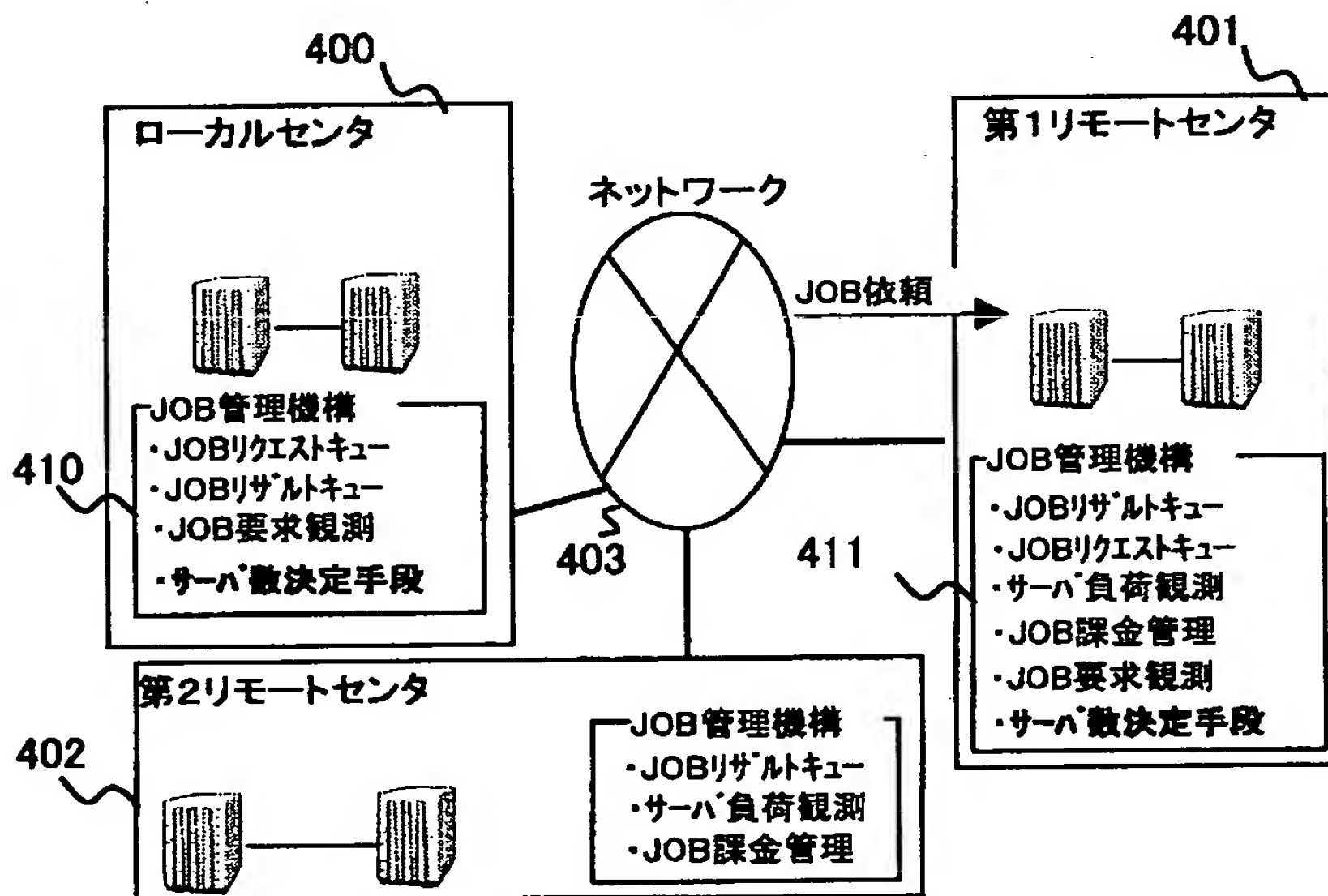
【図 3】

図3



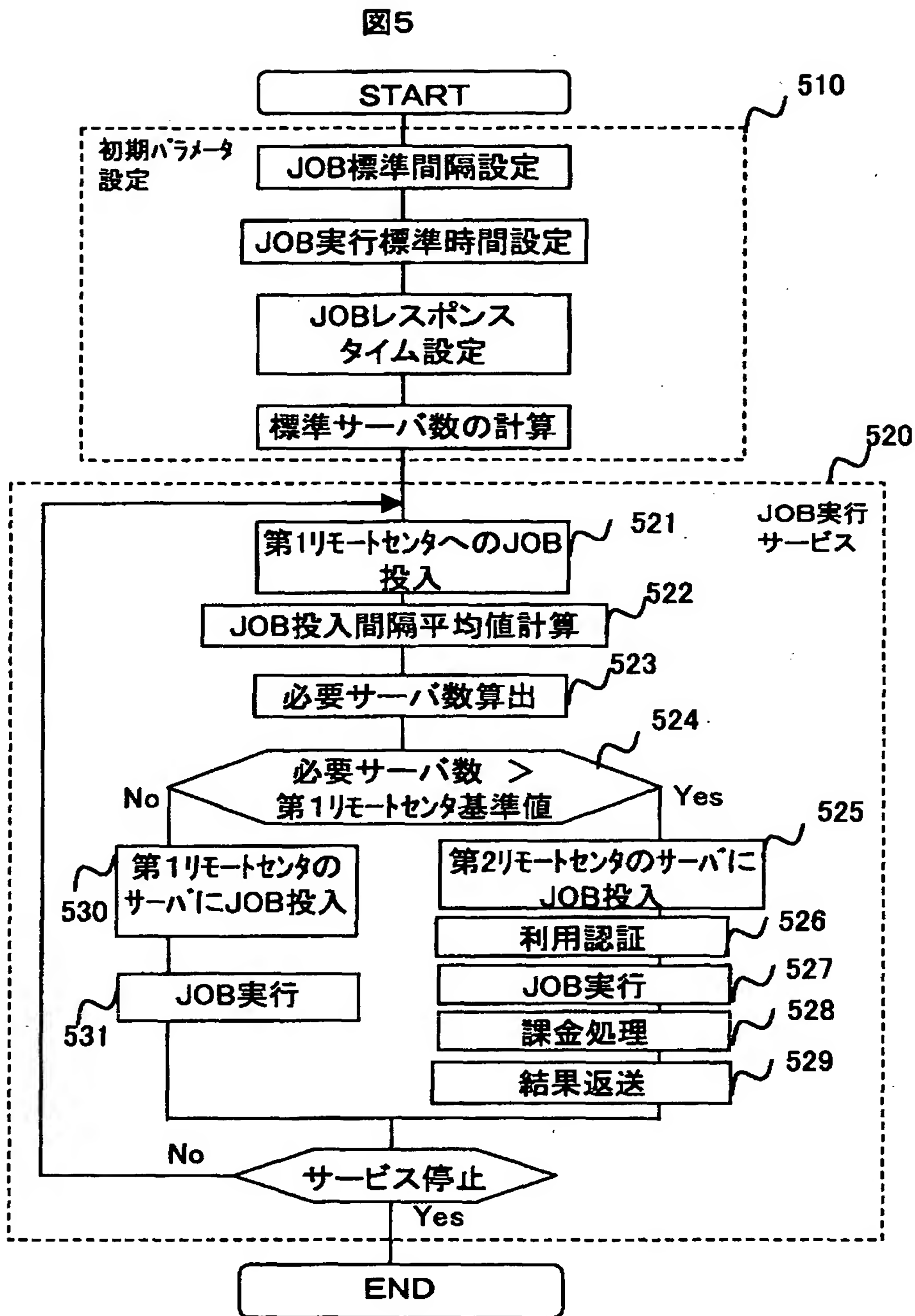
【図 4】

図4



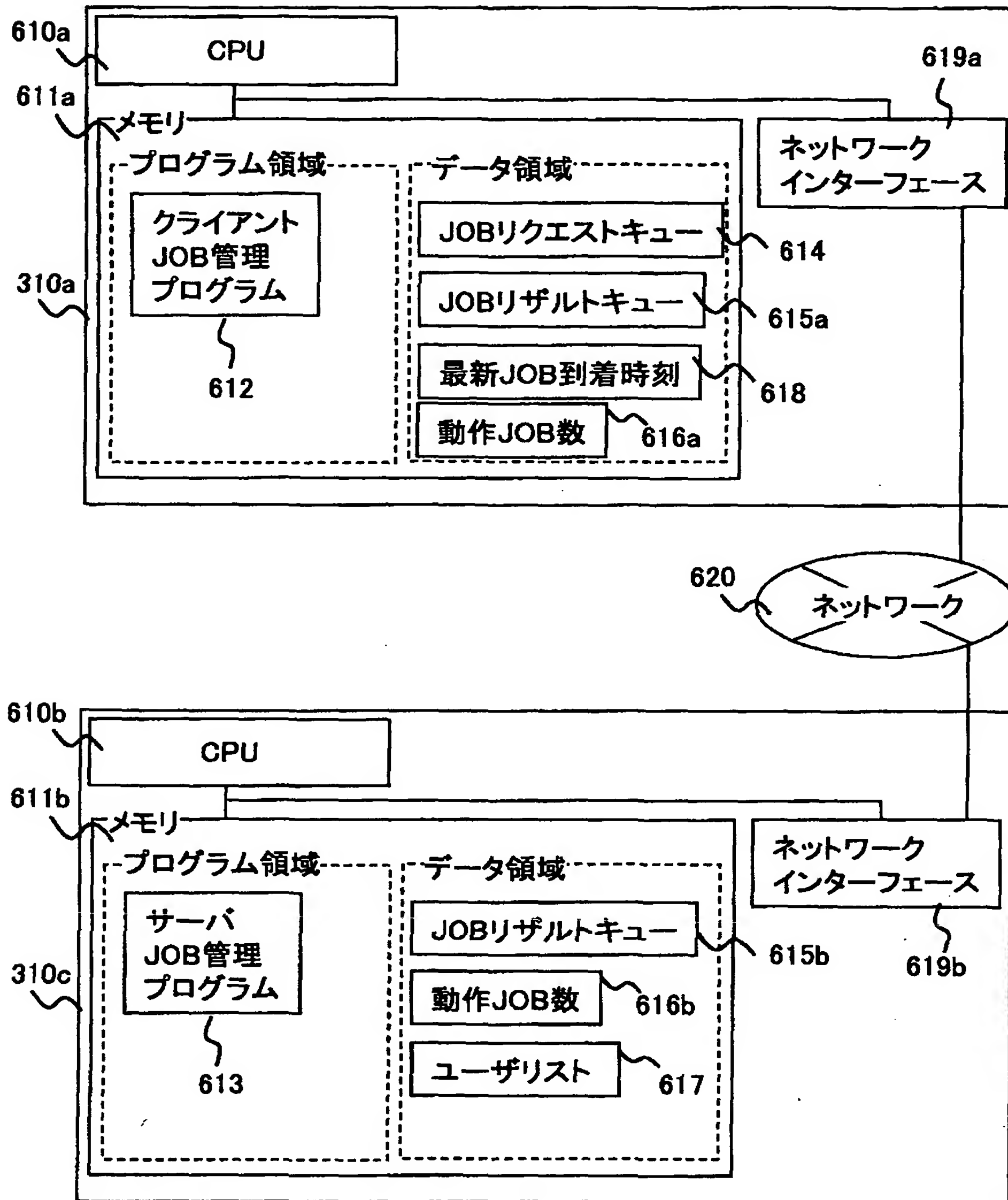


【図 5】

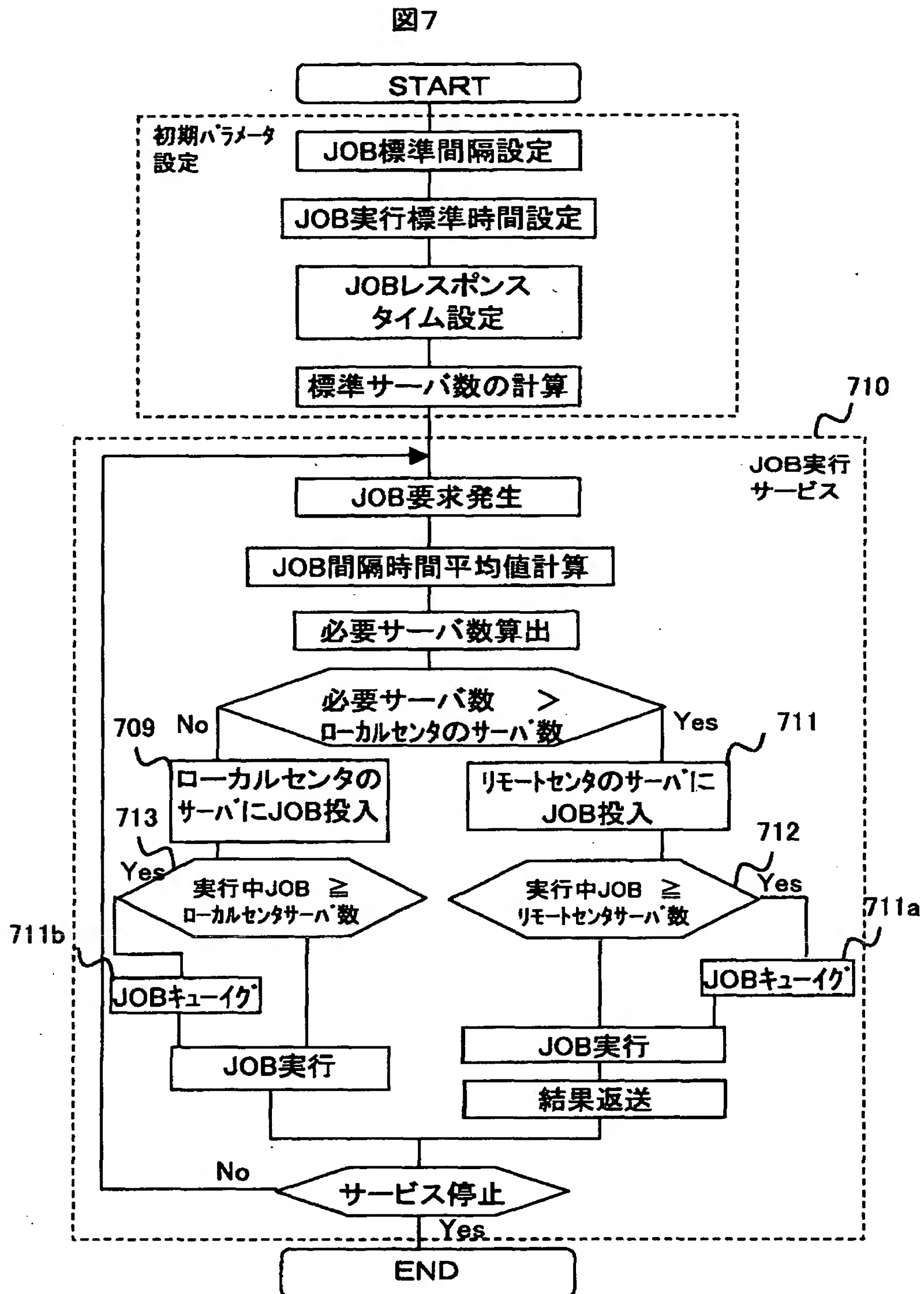


【図6】

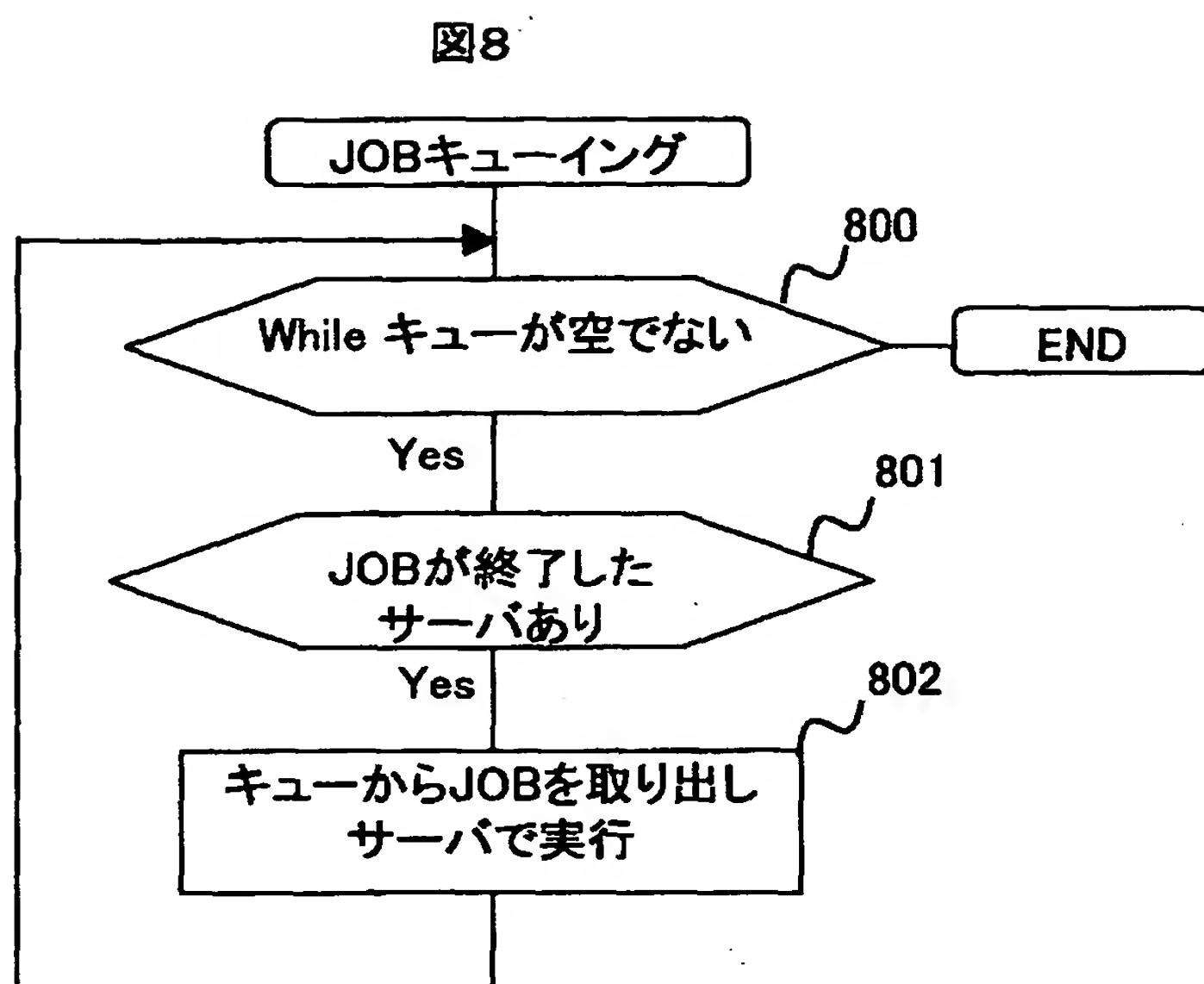
図6



【図 7】

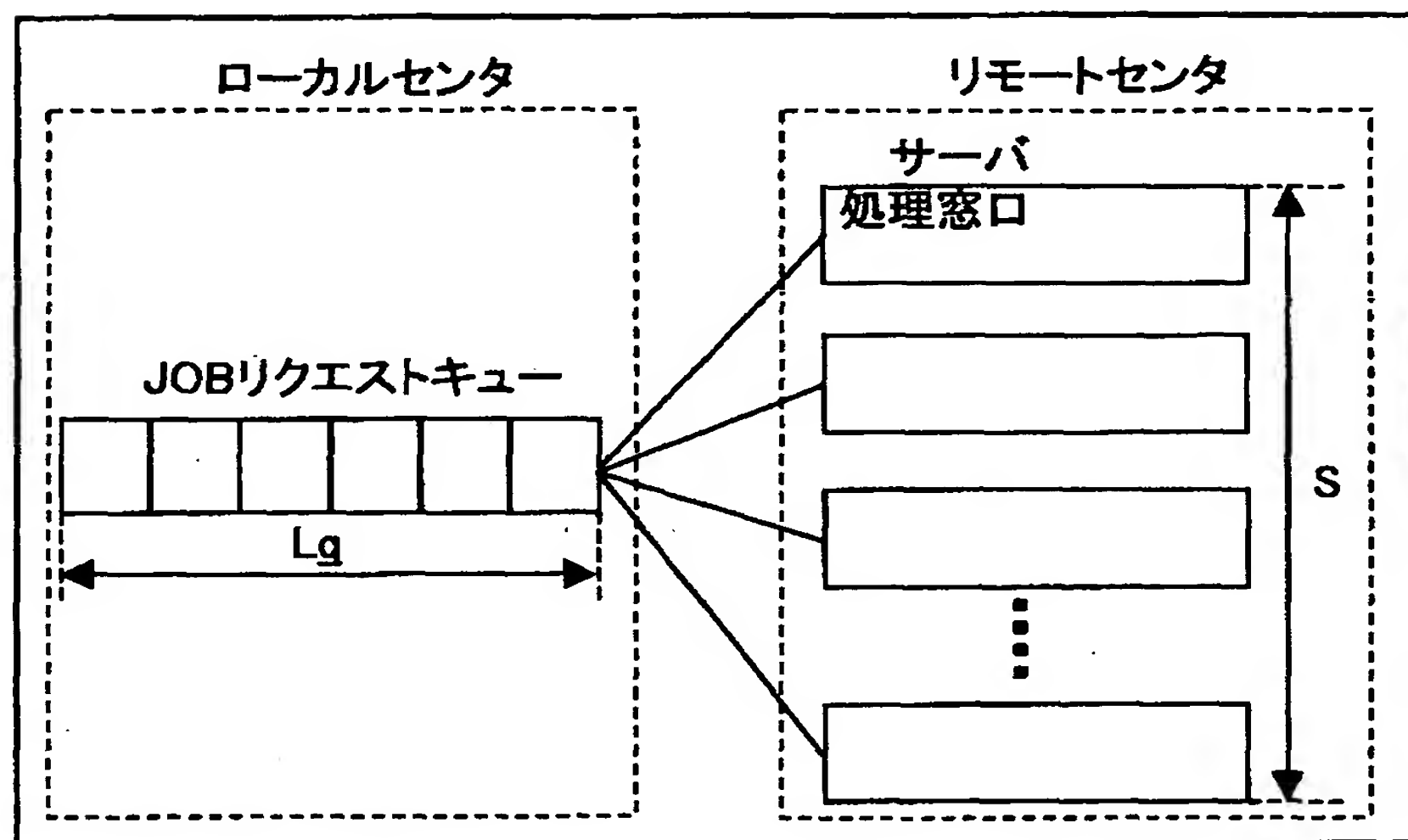


【図 8】



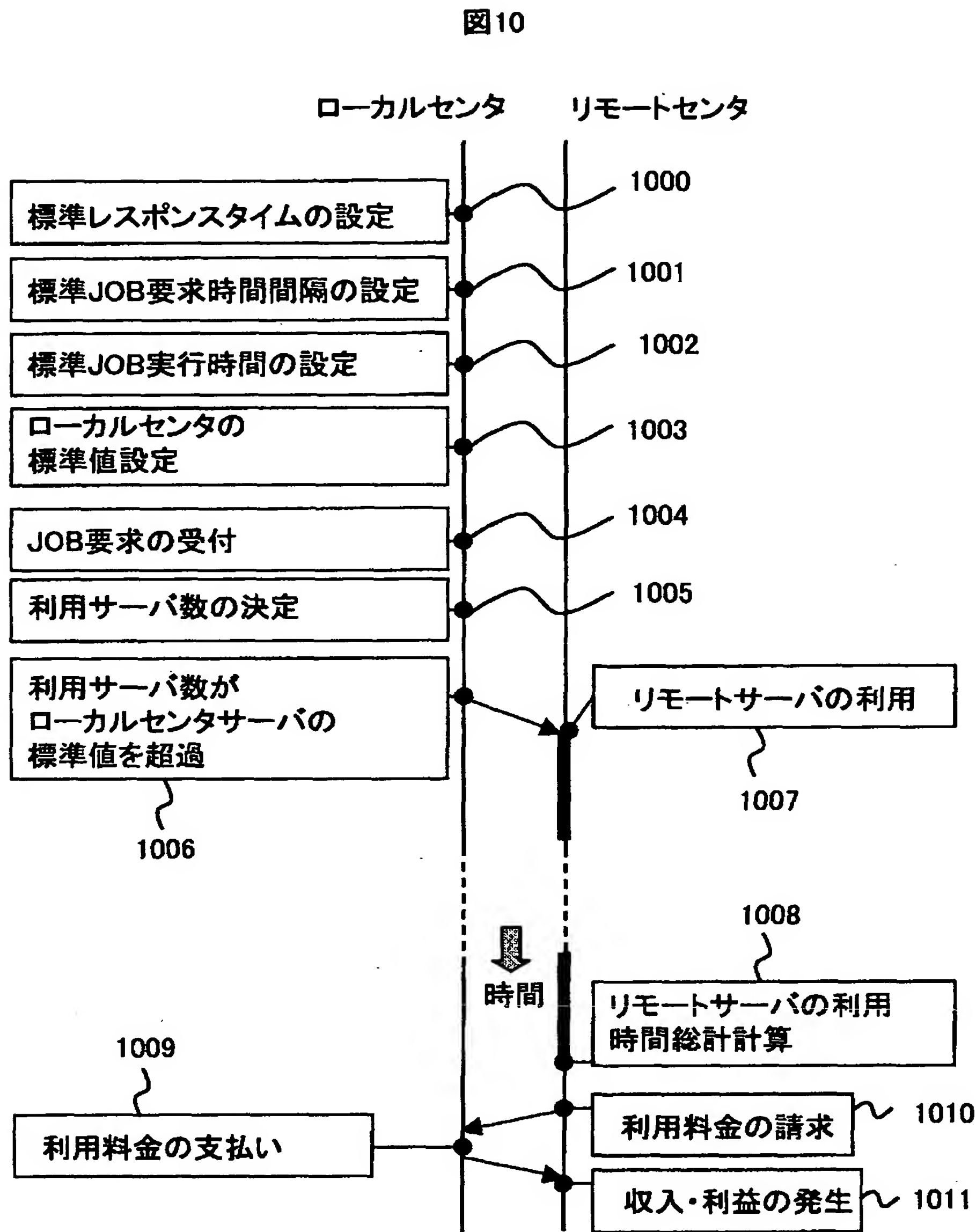
【図 9】

図9



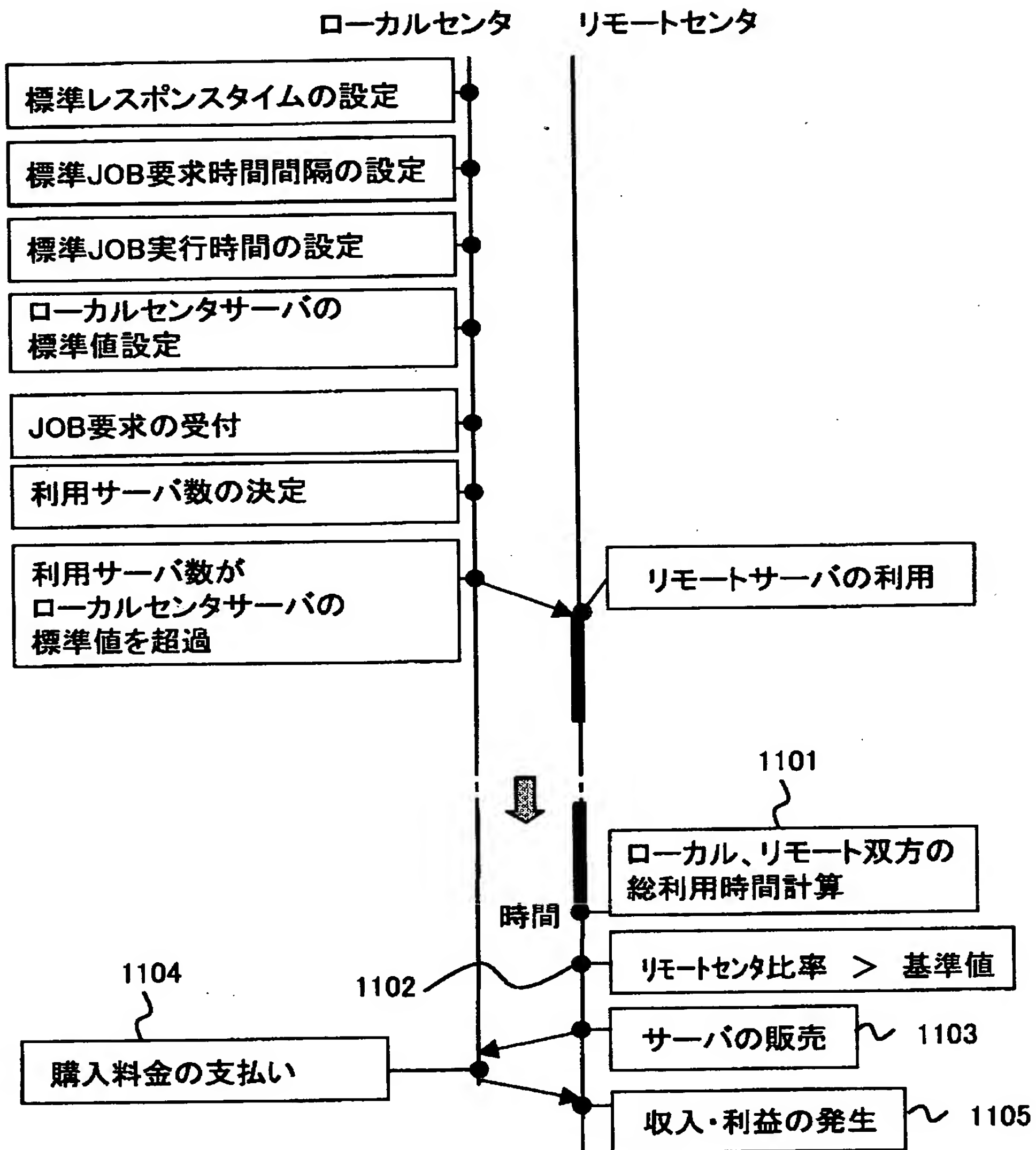
ローカルセンタ、リモートセンタの待ち行列モデル

【図 1 0】



【図 1 1】

図11





【書類名】                    要約書

【要約】

【課題】    グリッドのように、ネットワークで結合された複数の計算機センタにおいて、適切な J O B の負荷分散を実現する。

【解決手段】    J O B 要求の平均的な間隔と J O B 実行時間の平均時間と要求されるレスポンスタイムから必要なサーバ数を決定する手段を設けることで、ネットワークの遅延がある環境でも、サーバ数が不足した際にネットワークで結合された遠方の計算機センタへ均等な負荷の J O B を投入する。

【効果】    ネットワークの遅延が存在しても、遠方の計算機センタの負荷を観測しないために、正しい負荷分散が可能となる。

【選択図】    図 1

特 2 0 0 3 - 1 4 1 2 9 2

## 認定・付加情報

特許出願の番号	特願 2 0 0 3 - 1 4 1 2 9 2
受付番号	5 0 3 0 0 8 3 1 3 8 9
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 5 年 5 月 2 1 日

### < 認定情報・付加情報 >

【提出日】 平成15年 5月20日

次頁無

出 願 人 履 歴 情 報

識別番号 [ 0 0 0 0 0 5 1 0 8 ]

1. 変更年月日 1 9 9 0 年 8 月 3 1 日

[変更理由] 新規登録

住 所 東京都千代田区神田駿河台 4 丁目 6 番地

氏 名 株式会社日立製作所